I CLAIM:

CLAIMS

1. An apparatus comprising:

at least one processor;

a memory coupled to the at least one processor;

a user-extensible object oriented framework residing in the memory, the framework including at least one core function that cannot be modified by a user and at least one extensible function defined by a user to customize the framework and thereby define a desired information retrieval system, the framework including:

a load document processor that loads and preprocesses a plurality of documents;

an index processor that creates at least one word index corresponding to the plurality of documents; and

a query processor that receives a query and determines if any of the plurality of documents match the query by processing the query and comparing the processed query to the plurality of words in the at least one word index, thereby providing a query result.

2. The apparatus of claim 1 wherein the index processor creates at least one word index in response to a build index request from a user.

3. The apparatus of claim 1 wherein the framework further includes:

a frequency counter that indicates the number of times a word appears in the at least one word index.

4. The apparatus of claim 1 wherein the framework further includes:

a table that maps a word index to the indexed document from which it was preprocessed.

5. The apparatus of claim 1 wherein the preprocessing by the load document processor includes a parsing method that identifies text words from other text characters.

6. The apparatus of claim 1 wherein the preprocessing by the load document processor includes a stoplist method that 1) identifies text words not containing sufficient information to be useful in providing a query result and 2) deletes such text words.

7. The apparatus of claim 1 wherein the preprocessing by the load document processor includes a stemming method that 1) identifies text word stems of which a text word is a formative, and 2) replaces the text word with the stem.

8. A program product comprising:

(A) a user-extensible object oriented framework mechanism comprising:

(1) a load document processor that loads and preprocesses a plurality of documents;

(2) an index processor that creates at least one word index corresponding to the plurality of documents; and

(3) a query processor that receives a query and determines if any of the plurality of documents match the query by processing the query and comparing the processed query to the plurality of words in the at least one word index, thereby providing a query result; and

(B) computer-readable signal bearing media bearing the framework mechanism.

9. The program product of claim 8 wherein the computer-readable signal bearing media comprises recordable media.

10. The program product of claim 8 wherein the computer-readable signal bearing media comprises transmission media.

11. The program product of claim 8 wherein the index processor creates at least one word index in response to a build index request from a user.

12. The program product of claim 8 wherein the framework mechanism further includes:

a frequency counter that indicates the number of times a word appears in the at least one word index.

13. The program product of claim 8 wherein the framework mechanism further includes:

a table that maps a word index to the indexed document from which it was preprocessed.

14. The program product of claim 8 wherein the preprocessing by the load document processor includes a parsing method that identifies text words from other text characters.

15. The program product of claim 8 wherein the preprocessing by the load document processor includes a stoplist method that 1) identifies text words not containing sufficient information to be useful in providing a query result and 2) deletes such text words.

16. The program product of claim 8 wherein the preprocessing by the load document processor includes a stemming method that 1) identifies text word stems of which a text word is a formative, and 2) replaces the text word with the stem.

17. A method of retrieving information from a plurality of documents comprising the steps of:

(1) providing a user-extensible object oriented framework mechanism;

(2) extending the object oriented framework mechanism; and

(3) executing the extended object oriented framework mechanism, the executing framework mechanism performing the steps of:

(A) loading and preprocessing a plurality of documents;

(B) creating at least one word index corresponding to the plurality of documents; and

(C) receiving a query and determining if any of the plurality of documents match the query by processing the query and comparing the processed query to the plurality of words in the at least one word index, thereby providing a query result.

18. The method of claim 17 wherein the framework mechanism performs step (B) in response to a build index request from a user.

19. The method of claim 17 wherein the executing framework mechanism further preforms the step of counting the number of times a word appears in the at least one word index.

20. The method of claim 17 wherein the executing framework mechanism further preforms the step of mapping a word index to the indexed document from which it was preprocessed.

21. The method of claim 17 wherein the preprocessing of a document includes the step of identifying text words from other text characters.

22. The method of claim 17 wherein the preprocessing of a document includes the steps of:

1) identifying text words not containing sufficient information to be useful in providing a query result; and

2) deleting such text words.

23. The method of claim 17 wherein the preprocessing of a document includes the steps of:

1) identifying text word stems of which a text word is a formative; and

2) replacing the text word with the stem.

* * *